

An Efficient Search Strategy for Aggregation and Discretization of Attributes of Bayesian Networks Using Minimum Description Length

J.N. Corcoran¹, D. Tran², and N.D. Levine³

¹University of Colorado, Boulder

²University of Kansas

³Naval Postgraduate School

April 4, 2014

Abstract

Bayesian networks are convenient graphical expressions for high dimensional probability distributions representing complex relationships between a large number of random variables. They have been employed extensively in areas such as bioinformatics, artificial intelligence, diagnosis, and risk management. The recovery of the structure of a network from data is of prime importance for the purposes of modeling, analysis, and prediction. Most recovery algorithms in the literature assume either discrete or continuous but Gaussian data. For general continuous data, discretization is usually employed but often destroys the very structure one is out to recover. Friedman and Goldszmidt [11] suggest an approach based on the minimum description length principle that chooses a discretization which preserves the information in the original data set, however it is one which is difficult, if not impossible, to implement for even moderately sized networks. In this paper we provide an extremely efficient search strategy which allows one to use the Friedman and Goldszmidt discretization in practice.

*Postal Address: J.N. Corcoran, Department of Applied Mathematics, University of Colorado, Box 526
Boulder CO 80309-0526, USA; email: corcoran@colorado.edu; phone: 303-492-0685
Keywords: Bayesian networks, discretization, minimum description length
AMS Subject classification: 62-09, 68P30, 62C99

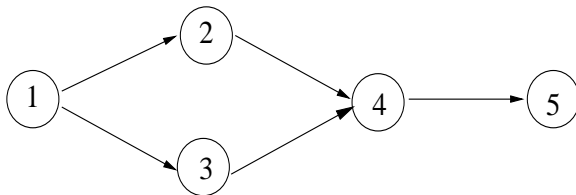
1 Introduction

Bayesian networks are convenient graphical expressions for high dimensional probability distributions representing complex relationships between a large number of random variables. They have been employed extensively in areas such as bioinformatics ([4], [12], [17], [16]), artificial intelligence ([15], [13], [21]), diagnosis ([8], [18], [23]), and risk management ([7], [10], [22]) for the purposes of general modeling, prediction, and diagnosis.

A Bayesian network consists of a directed acyclic graph (DAG), in which nodes represent random variables, and a set of conditional probability distributions. For example, the graph in Figure 1 represents a network for 5 variables, X_1, X_2, X_3, X_4 , and X_5 with a joint probability density $p(x_1, x_2, x_3, x_4, x_5)$ which is assumed to factor as

$$p(x_1, x_2, x_3, x_4, x_5) = p(x_1) \cdot p(x_2|x_1) \cdot p(x_3|x_1) \cdot p(x_4|x_2, x_3) \cdot p(x_5|x_4).$$

Figure 1: A Directed Acyclic Graph



In general, for a Bayesian network on n nodes, the joint density for the corresponding n random variables may be written as

$$p(x_1, x_2, \dots, x_n) = \prod_{i=1}^n p(x_i|\Pi_i) \tag{1}$$

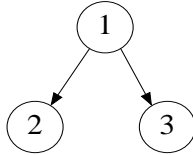
where Π_i is used to denote the set of random variables corresponding to “parent” nodes for node i . For the example in Figure 1, we have $\Pi_1 = \emptyset$, $\Pi_2 = \{x_1\}$, $\Pi_3 = \{x_1\}$, $\Pi_4 = \{x_2, x_3\}$, and $\Pi_5 = \{x_4\}$.

Given m observations of (X_1, X_2, \dots, X_n) , our ultimate goal is to infer the connecting arrows for a corresponding n -node graph. There are many ways to do this rather successfully for discrete data or for Gaussian data. In the case of non-Gaussian continuous data, discretization is usually employed, however care must be taken not to destroy the very structure one wishes to retrieve. Since the problem involves mapping entire intervals of values to discrete values, it is closely related to the “aggregation problem” where one reduces the number of

possible values taken on by a node by mapping collections of values in already discrete data to single points. (We will think of this as a “discretization of discrete data” and will still refer to it as discretization.)

For an example of a discretization masking important attributes of the underlying network structure, consider the three node DAG show in Figure 2 where the corresponding random variables X_1 , X_2 , and X_3 each take on values in $\{1, 2, 3\}$. Consider the discretization to

Figure 2: A Three Node DAG



random variables X_1^* , X_2^* , and X_3^* defined as $X_i^* = 1$ if $X_i \in \{1, 2\}$ and $X_i^* = 2$ if $X_i = 3$.

The original network structure implies that the variables represented by nodes 2 and 3 are conditionally independent given the value of the variable at node 1, yet it is easy to verify that we have, after discretization,

$$P(X_2^* = 1, X_3^* = 1 | X_1^* = 1) \neq P(X_2^* = 1 | X_1^* = 1) \cdot P(X_3^* = 1 | X_1^* = 1),$$

for example. Thus, there is little hope of recovering the correct network structure after this discretization with data that have been discretized in this way.

Friedman and Goldszmidt [11] suggest an approach based on the *minimum description length principle* that chooses a discretization which preserves the information in the original data set. However, their approach, which we describe in Section 3, requires that we compute and compare scores for every different possible discretization of the data. The number of possibilities is often prohibitively large.

In this paper we give a search strategy in which only a very small subset of minimum description length scores must be computed. Under certain simplifying assumptions, we prove that the procedure will identify the optimal score. Furthermore, in the case where there is a “correct” discretization in the sense that distributions of nodes are unchanged over entire intervals of values for children, we show that the optimal scoring network does in fact correspond to the data generating network.

2 Likelihood-Based Recovery of Bayesian Networks for Discrete Data

We will use this section to establish some notation that will be used throughout this paper and to briefly review commonly used likelihood-based methods for recovering the structure of Bayesian networks.

Consider a Bayesian network on n nodes. For $i = 1, 2, \dots, n$, let X_i denote the random variable associated with node i . We will refer to node i and the associated random variable X_i , interchangeably. Suppose that X_i can take on values in $\{1, 2, \dots, ||X_i||\}$. Let Π_i denote the set of parent node indices for node i , let $|\Pi_i|$ denote the number of parents for node i , and let $||\Pi_i||$ denote the number of possible parent configurations. For example, for the network in Figure 1,

$$|\Pi_4| = 2 \quad \text{and} \quad ||\Pi_4|| = ||X_2|| \cdot ||X_3||.$$

For $i = 1, 2, \dots, n$, $j = 1, 2, \dots, ||\Pi_i||$, and $k = 1, 2, \dots, ||X_i||$, define parameters

$$\theta_{ijk} := P(X_i = k | \Pi_i = j).$$

Here, “ $\Pi_i = j$ ” refers to a specific set of values for the parent variables, configurations of which have been enumerated and are now indexed by j .

Now, from the joint density (1), the likelihood for the entire $m \times n$ dimensional data set may be written as

$$L(\theta) := \prod_{i,j,k} \theta_{ijk}^{n_{ijk}}$$

where n_{ijk} is the total number of times in the sample that X_i is observed to have value k when its parents take on configuration j . (To keep this concise expression for the likelihood, we define, for a parentless node i , $|\Pi_i| = 1$ and $\theta_{i1k} = P(X_i = k)$.)

There are many ways to recover networks from data. Indeed, we may not even want to think in terms of “one best network” and instead use a model averaging approach or one that constructs a best network by combining best “features” (for example edges) from several networks. In this paper, we will restrict our attention to simple methods for recovering a single “best” network as measured by various standard likelihood and information criterion indices. We will assume a manageable number of networks to score. Of course, the number of possible networks increases superexponentially in the number of nodes— the results of this paper might then be applied using Monte Carlo search strategies.

1. Log-Likelihood (LL)

Given m n -tuples of data points, we compute the log-likelihood for every possible graph. For each DAG, we have a different set of relevant θ parameters. Given a particular DAG, we estimate each θ with its maximum likelihood estimator

$$\hat{\theta}_{ijk} = \frac{\# \text{ observations with } X_i = k \text{ and } \Pi_i = j}{\# \text{ observations with } \Pi_i = j},$$

and then we report the log-likelihood

$$\ln L(\hat{\theta}) = \sum_{i,j,k} n_{ijk} \ln(\hat{\theta}_{ijk}).$$

As we can always increase the log-likelihood by including additional θ parameters, we expect the greatest log-likelihoods (“most likely models”) to coincide with DAGs with a maximal number of edges. Therefore, it is important include a penalty for overparameterized models. The two most common penalized log-likelihood statistics are given by the following information criteria.

2. Akaike’s Information Criterion (AIC)

Akaike’s Information Criterion (AIC) is, in its most general form, defined by

$$AIC = -2 \ln L + 2 \cdot (\#parameters).$$

Clearly, the goal is to minimize the AIC to ensure a good fitting model in the sense of maximizing the log-likelihood while penalizing for having too many parameters.

3. Bayesian Information Criterion (BIC)

The Bayesian Information Criterion (BIC) is defined by

$$BIC = -2 \ln L + (\#parameters) \cdot \ln(m),$$

where m is, as before, the sample size. As with the AIC, the goal is to minimize the BIC.

Both the AIC and BIC have rigorous justifications, from both Bayesian and frequentist points of view. AIC was derived from information theory though it can be thought of as Bayesian if one uses a clever choice of prior. On the other hand BIC, originally derived through Bayesian statistics as a measure of the Bayes factor, can also be derived as a non-Bayesian result. For more information on these widely used scoring criteria, we refer the

interested reader to [1],[2], [3], and [5] (AIC), and [14] and [20] (BIC). To make some broad generalizations, AIC, it can often overfit the model in terms of number of parameters. BIC, on the other hand, tends to overpenalize, or underfit the model. That being said, we have been most successful in recovering the correct structure of simulated Bayesian networks using AIC. Incidentally, the use of Bayesian networks does not necessarily imply a Bayesian approach to modeling where parameters are considered as random variables. In this paper, we are taking the frequentist approach.

3 Minimum Description Length

Arguably, the point of statistical modeling is to find regularities in an observed data set. Discovered regularities allow the modeler to be able to describe the data more succinctly. The minimum description length (MDL) principle, introduced in 1978 by Jorma Rissanen [19], is a model selection technique that chooses, as the best model, the one that permits the shortest encoding of both the model and the observed data.

In this section, we describe how Friedman and Goldszmidt [11] define a description length that can be used for network recovery from discrete data. It is the approximate length of storage space, measured in bits, for the binary representation of the DAG (network structure), the parameters (the θ_{ijk}) that, together with the DAG, define a Bayesian network, and the data itself. We will see, in the end, that it is simply another penalized likelihood approach.

In what follows, we repeatedly use the fact that an integer k can be encoded in approximately $\lceil \log_2 k \rceil$ bits. For example, the decimal value 9 becomes, in binary, the $\lceil \log_2 9 \rceil = 4$ bit number 1001. Throughout this paper, we will use \log to denote the base 2 logarithm, though, in the end the base is unimportant for the comparisons we will make.

Encoding the DAG

As in Section 2, we assume that X_i , can take on $\|X_i\|$ possible values, and that they are integers ranging from 1 to $\|X_i\|$. (For example, if X_1 can take on values in $\{2, 5, 11\}$, we would relabel the values as 1, 2, and 3.)

The approximate number of bits needed to encode the number of nodes/variables n , and

the number of possible values taken on by each of those variables is

$$\log n + \sum_{i=1}^n \log ||X_i||. \quad (2)$$

In order to completely describe the network structure, we must also include the number of parents for each node and the actual list of parents for each node. As a simplification, since the number of parents of node i , which is denote by $|\Pi_i|$, is always less than n , we can, conservatively, reserve $\log n$ bits to encode $|\Pi_i|$. Doing this for each node, we add

$$\sum_{i=1}^n \log n = n \log n \quad (3)$$

to our network description length.

For the actual list of parents, since the maximum value in the list of indices is n , we will use the conservative value of $\log n$ bits to encode each index. For node i , we must encode $|\Pi_i|$ different indices, each using a length of $\log n$ bits of space. So, in total, to encode all parent lists, we will use

$$\sum_{i=1}^n |\Pi_i| \cdot \log n \quad (4)$$

bits.

In total, we use

$$\log n + \sum_{i=1}^n \log ||X_i|| + \sum_{i=1}^n (1 + |\Pi_i|) \cdot \log n \quad (5)$$

bits to encode the DAG structure.

Encoding the Parameters

The Bayesian network consists of a DAG together with a collection of parameters $\theta_{ijk} = P(X_i = k | \Pi_i = j)$. Since $\sum_k \theta_{ijk} = 1$, we only need to encode $||\Pi_i|| \cdot (||X_i|| - 1)$ parameters for node i . However, the parameters are not integer valued. In our case, for network recovery, we will actually be storing/encoding parameters that have been estimated from our m n -dimensional data points. Friedman and Goldszmidt [11] indicate that the “usual choice in the literature” is to use $\frac{1}{2} \log m$ bits per parameter. Thus, we will use

$$\frac{1}{2} \log m \cdot \sum_{i=1}^n ||\Pi_i|| \cdot (||X_i|| - 1) \quad (6)$$

bits to encode the estimated network parameters.

We add (5) and (6) in order to define description length for the Bayesian network. Since we are trying to infer the best connecting arrows for a graph on n nodes, we presumably already know the number of nodes and would not be encoding it. Thus, we drop the $\lceil \log n \rceil$ term and define the *network description length* as

$$DL_{net} = \sum_{i=1}^n \log \|X_i\| + \sum_{i=1}^n (1 + |\Pi_i|) \cdot \log n + \frac{1}{2} \log m \cdot \sum_{i=1}^n \|\Pi_i\| \cdot (\|X_i\| - 1). \quad (7)$$

Encoding the Data

Consider the string of digits 2213 stored, in binary as 1010111. (The binary representations of 1, 2, and 3 are 1, 10, and 11, respectively.) Without separators, the binary string 1010111 can not be decoded near the end. The 111 maybe be, in decimal, three 1's, a 1 followed by a 3, or a 3 followed by a 1. "Prefix codes" avoid this problem by encoding the digits with zeros and ones in a way so that no encoded digit is a prefix of any other encoded digit. Further consideration can be made to ensure the code is not only a prefix code, but one that results in the maximum compression of the data by assigning, to the original decimal digits, binary codes of lengths inversely proportional to the frequencies with which they appear in the original data.

In the context of Bayesian networks, we wish to encode the values in the $(m \times n)$ -dimensional data set with code lengths that are inversely proportional to the frequencies (equivalently, estimated probabilities) with which they appear in the data. Friedman and Goldszmidt [11] use *Shannon coding* [6] which is not optimal in terms of compression, but which encodes each n -dimensional data point $\vec{x}_i = (x_{i1}, x_{i2}, \dots, x_{in})$ using approximately $-\log p(\vec{x}_i)$ bits where $p(\vec{x}_i)$ is the joint density for X_1, X_2, \dots, X_n evaluated at \vec{x}_i . Thus, the entire data set, consisting of m such vectors is encoded in approximately

$$DL_{data} = - \sum_{i=1}^m \log p(\vec{x}_i) \quad (8)$$

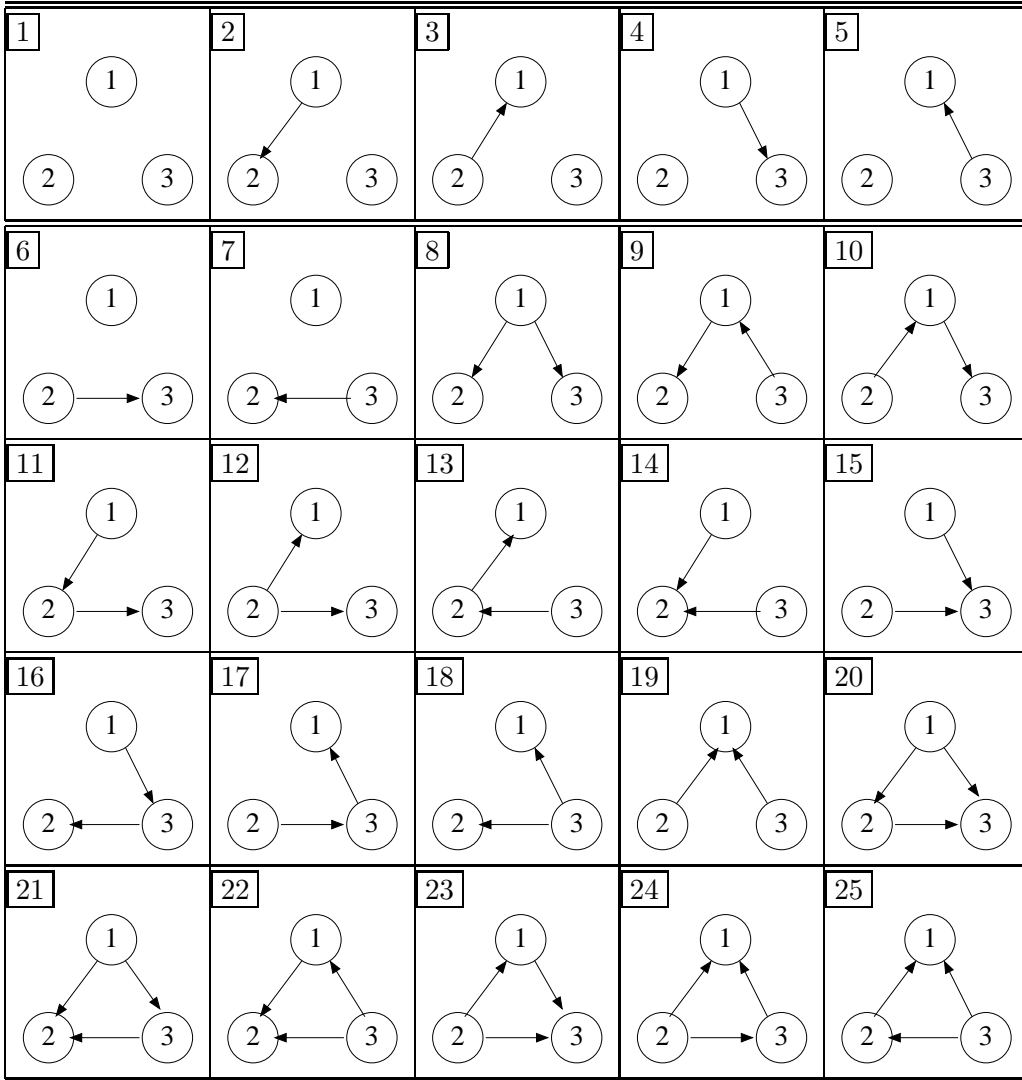
bits. This is desirable from a modeling standpoint since it corresponds to the familiar log-likelihood commonly used in statistical inference.

Summing (7) and (8), we define the description length for a Bayesian network and the observed data as

$$DL = \sum_{i=1}^n \log \|X_i\| + \sum_{i=1}^n (1 + |\Pi_i|) \cdot \log n + \frac{1}{2} \log m \cdot \sum_{i=1}^n \|\Pi_i\| \cdot (\|X_i\| - 1) - \sum_{i=1}^m \log p(\vec{x}_i). \quad (9)$$

Given discrete data and a collection of possible Bayesian networks, the network chosen by the MDL principle is the one that minimizes (9). In light of the form of DL_{data} , we see that this is simply a penalized log-likelihood scoring metric, similar to the AIC and BIC discussed in Section 2. We now illustrate the performance of all three for a three node network. The list of all 25 DAGs corresponding to 3 node networks can be found in Table 1 and we will refer to these DAGs as they are numbered here.

Table 1: Directed Acyclic Graphs on Three Nodes



We fixed arbitrary parameters θ_{ijk} for DAG 8 in Table 1 and simulated 100,000 values of (X_1, X_2, X_3) . (Each X_i was assumed to take on values in $\{1, 2, \dots, 6\}$.)

Results are shown in Table 2 along with the previously computed values of AIC and BIC. The AIC, BIC, and MDL scores all recovered the correct network “up to Markov equivalence”. Recovering the specific graph within a Markov equivalence class (a causality problem) requires experimental as opposed to simply observed data and is not the subject of this paper.

4 Minimum Description Length for Discretization

A discretization of data for the random variable X_i , represented by node i , is a mapping from the range of values in the data set to the set $\{1, 2, \dots, k_i\}$ for some $k_i \geq 1$. It can be described by ordering the distinct values observed for X_i and inserting up to $k_i - 1$ “thresholds”. For example

$$\underbrace{0.38 \quad 0.42 \quad 0.53 \quad 0.71}_{\text{map to 1}} \mid \underbrace{1.37 \quad 1.94 \quad 2.10}_{\text{map to 2}} \mid \underbrace{5.38 \quad 7.11}_{\text{map to 3}}. \quad (10)$$

In this paper, we will assume that only node i needs to be discretized and that the remaining nodes are discrete or have already been discretized. We refer the reader to Friedman and Goldszmidt [11] for a discussion of multiple node discretization which essentially involves fixing discretizations for all but one node, discretizing that node, and repeating the process by cycling through all nodes repeatedly.

As before, we assume that we have m observations of the n -dimensional (X_1, X_2, \dots, X_n) . Let m_i be the number of distinct values taken on by X_i in the data set. Note that $m_i \leq m$, with equality possible only for truly continuous data. Define X_i^* as the discretized version of X_i . To discretize to $k_i \leq m_i$ values, we need to choose $k_i - 1$ thresholds to put in $m_i - 1$ spaces between ordered values. There are $\binom{m_i - 1}{k_i - 1}$ threshold configurations to consider. Since we will not know in advance how many values we should have in the discretized data set, we need to consider everything from $k_i = 1$, which corresponds to mapping all values for X_i in the data set to the single value of 1, to $k_i = m_i$, which corresponds to no discretization at all. In total, there are

$$\binom{m_i - 1}{0} + \binom{m_i - 1}{1} + \dots + \binom{m_i - 1}{m_i - 1} = 2^{m_i - 1}$$

A set of DAGs with the same set of edges are said to be *Markov equivalent* if they share the same set of conditional independence relations among variables.

Table 2: AIC, BIC and MDL Recovery

Graph Number (n)	AIC	BIC	MDL
1	1040886.085	1041057.317	520443.714
2	1001501.922	1001958.543	500885.425
3	1001501.961	1001958.543	500885.425
4	991255.256	991711.879	495762.093
5	991255.259	991711.879	495762.093
6	1036182.434	1036639.054	518225.681
7	1036182.434	1036639.054	518225.681
8	951871.096	952613.104	476213.804
9	951871.096	952613.104	476213.804
10	951871.096	952613.104	476213.804
11	996798.271	997540.279	498677.392
12	996798.271	997540.279	498677.392
13	996798.271	997540.279	498677.392
14	991435.745	936604.692	496536.904
15	986551.608	987293.616	493554.060
16	986551.608	987293.616	493554.060
17	986551.608	987293.616	493554.060
18	956755.233	958924.180	479196.648
19	952051.582	954505.917	476988.615
20	952051.582	954505.917	476988.615
21	952051.582	954505.917	476988.615
22	952051.582	954505.917	476988.615
23	952051.582	954505.917	476988.615
24	952051.582	954505.917	476988.615
25	952051.582	954505.917	476988.615

discretizations to consider.

Friedman and Goldszmidt [11] define a description length score for a network with X_i discretized into a particular configuration of k_i thresholds using essentially four terms.

These terms include DL_{net} and DL_{data} , previously described in (7) and (8), computed now after discretization— we will call these terms DL_{net}^* and DL_{data}^* . Also included are terms that encode the index denoting a particular discretization policy and description length for information needed to recover the original data from the discretized data.

Encoding the Discretization Policy

For fixed m_i and k_i , there are $\binom{m_i - 1}{k_i - 1}$ different possible configurations for thresholds. Assume we have labeled them from 1 to $\binom{m_i - 1}{k_i - 1}$. Storing the index for a particular policy will take at most $\left\lceil \log \binom{m_i - 1}{k_i - 1} \right\rceil$ bits. Friedman and Goldszmidt use a conservative upper bound based on the inequality

$$\binom{n}{k} \leq 2^{nH(k/n)},$$

where

$$H(p) := -p \log p - (1 - p) \log(1 - p),$$

and conservatively reserve

$$DL_{DP} = (m_i - 1)H\left(\frac{k_i - 1}{m_i - 1}\right) \quad (11)$$

to encode the discretization policy. (In practice, we define $H(0) = H(1) = 0$.)

Encoding Recovery of Original Data

Consider again the example of 9 ($m_i = 9$) distinct values for X_i given by (10) from a data set with $m \geq m_i$ values. Every time we assign a discretized value to an original value, we should store the original value for recovery. Instead, however, we will store the Shannon binary code for the original value using estimated conditional probabilities based on the entire data set. For example, the value 2.10 might appear in the entire data set one-fourth of the time. That is, $\hat{P}(X_i = 2.10) = 1/4$. Among the instances of 1.37, 1.94, and 2.10, it might appear half of the time. That is, $\hat{P}(X_i = 2.10 | X_i^* = 2) = 1/2$. Given a discretized value of $X_i^* = 2$, we will encode the original value, X_i , using approximately $-\log \hat{P}(X_i | X_i^* = 2)$ bits. In

total, for recovering original data from discretized data, we add

$$DL_{rec} = - \sum_{i=1}^m \log \hat{P}(X_i | X_i^*) \quad (12)$$

to the description length.

In summary, Friedman and Goldszmidt [11] define the *description length discretization score* as

$$DL^* = (m_i - 1)H\left(\frac{k_i - 1}{m_i - 1}\right) + DL_{net}^* + DL_{data}^* + DL_{rec}. \quad (13)$$

Given a data set and a particular network structure, one scores various discretization mappings for X_i and chooses the discretization that minimizes (13). For a given network, the score in (13) will change over discretizations only in terms directly linked to the i th node. Thus, as Friedman and Goldszmidt point out, we only need to consider the “local description length score” defined, by picking out relevant terms, as

$$\begin{aligned} DL_{local} = & (m_i - 1)H\left(\frac{k_i - 1}{m_i - 1}\right) + \log k_i \\ & + \frac{1}{2} \log m \left[|\Pi_i| (k_i - 1) + \sum_{j: X_i \in \Pi_j} |\Pi_j^*| (|X_j| - 1) \right] \\ & - m \left[\hat{I}(X_i^*, \Pi_i) + \sum_{j: X_i \in \Pi_j} \hat{I}(X_j, \Pi_j^*) \right]. \end{aligned} \quad (14)$$

Here, Π_j^* is the set of parents for node j , denoted with an asterisk since it includes the discretized X_i^* , and,

$$\hat{I}(\vec{X}, \vec{Y}) = \sum_{\vec{x}, \vec{y}} \hat{P}(\vec{X} = \vec{x}, \vec{Y} = \vec{y}) \cdot \log \left(\frac{\hat{P}(\vec{X} = \vec{x}, \vec{Y} = \vec{y})}{\hat{P}(\vec{X} = \vec{x}) \hat{P}(\vec{Y} = \vec{y})} \right) \quad (15)$$

is the estimated mutual information between random vectors \vec{X} and \vec{Y} . ($\hat{I}(\vec{X}, \emptyset)$ is defined to be zero.)

5 Minimizing the Local Description Length Score

Computation of a single value of (14), which is based on network structure through the information terms, can be quite time consuming. For even moderately sized m_i , computation of (14) repeatedly to check all 2^{m_i-1} discretization policies can be prohibitive, and when multiple/all nodes need to be discretized (a procedure that involves several passes through

each node as discussed in [11]), computation of (14) becomes almost impossible. Friedman and Goldszmidt [11] give some further computational simplifications and suggest a greedy search routine.

The purpose of this paper is to provide an alternative efficient search strategy for the smallest DL_{local} score. We assert that, for a single node discretization, one need only check m_i values of DL_{local} as opposed to 2^{m_i-1} . Multiple continuous nodes can then be cycled for discretization just as Friedman and Goldszmidt have suggested.

For notational simplicity, we assume, for the remainder of this paper, that the node to be discretized is labeled as node 1. Also, as we are comparing values of DL_{local} for various discretizations of X_1 , we will drop all asterisk superscript notation, as it is understood that we are considering discretized values.

Single Threshold Top-Down Search Strategy

Let $DL_{local}(0)$ be (14) with all $m_1 - 1$ thresholds in place. For $j = 1, 2, \dots, m_1 - 1$, let $DL_{local}(-j)$ be (14) with all thresholds except for the j th threshold.

In order to minimize DL_{local} over all 2^{m_1-1} discretization policies for X_1 , make comparisons of $DL_{local}(0)$ with $DL_{local}(-j)$ for $j = 1, 2, \dots, m_1 - 1$. If $DL_{local}(-j) \leq DL_{local}(0)$, remove the j th threshold.

We first consider the effectiveness of this search strategy in the very ideal situation where we augment given discrete data by introducing superfluous values for node 1 in a larger discrete set. For example, we might replace values of 5 in the original data set with values in $\{5, 6, 7\}$ with some arbitrary probabilities, whereupon we hope that our search strategy will minimize DL_{local} and that the configuration of thresholds that does such will correctly map values in $\{5, 6, 7\}$ back to the original value of 5. We will prove, in this case, that the “single threshold top-down” search strategy will find the discretization policy that minimizes DL_{local} among all 2^{m_1-1} discretizations for a large enough sample size m .

Indeed, m , m_1 , $\|\Pi_1\|$, and $\|X_j\|$ are constant, and $0 \leq H(p) \leq 1$. Note that

$$(m_1 - 1)H\left(\frac{k_1 - 1}{m_1 - 1}\right) + \log k_1 + \frac{1}{2} \log m \left[\|\Pi_1\|(k_1 - 1) + \sum_{j: X_1 \in \Pi_j} \|\Pi_j\|(\|X_j\| - 1) \right] \quad (16)$$

will be constant over any discretization policy for X_1 with a fixed number of thresholds. In particular, when comparing all possible single threshold removals, starting with any fixed

number of thresholds, we can restrict our attention to maximizing

$$\hat{I}(X_1, \Pi_1) + \sum_{j: X_1 \in \Pi_j} \hat{I}(X_j, \Pi_j). \quad (17)$$

In the next Section, we consider what our search strategy does to (17) in the absence of (16) in an ideal situation where there is a “correct” discretization. We will see that it maximizes (17) and that it does so while leaving a minimal number of thresholds. In Section 5.2, we will consider (16) and conclude that we are indeed minimizing DL_{local} . We will also see that it finds the correct discretization.

5.1 A Closer look at Information Terms in an Ideal Situation

As the notation to follow gets a bit cumbersome in the general case, we illustrate the proof of most claims in this Section and the next with a concrete example. Consider a two-node network where node 1 is a parent to node 2, and assume that both nodes take on values in $\{1, 2, 3\}$. From m data points, we can produce estimates

$$\begin{aligned} \hat{p}(i, j) &:= \hat{P}(X_1 = i, X_2 = j) \\ \hat{p}_1(i) &:= \hat{P}(X_1 = i), \text{ and} \\ \hat{p}_2(j) &:= \hat{P}(X_2 = j), \end{aligned}$$

for $i, j \in \{1, 2, 3\}$.

In general, for this two node network, we would assume that node 1 takes on values in $\{1, 2, \dots, m_1\}$, node 2 takes on values in $\{1, 2, \dots, m_2\}$, and one would work with the estimates $\hat{p}(i, j)$, $\hat{p}_1(i)$, and $\hat{p}_2(j)$ for $i \in \{1, 2, \dots, m_1\}$ and $j \in \{1, 2, \dots, m_2\}$.

We now “explode” the data for our specific example at node 1 into values in $\{1, 2, 3, 4, 5, 6\}$ by replacing instances of 1 with values in $\{1, 2\}$ with probabilities $1/3$ and $2/3$, respectively, replacing original instances of 2 with values in $\{3, 4, 5\}$ with probabilities $2/7$, $4/7$, and $1/7$, respectively, and replacing original instances of 3 with the value 6. Thus, in our “exploded data set”, node 1 is taking on values in $\{1, 2, 3, 4, 5, 6\}$ with probabilities denoted as $\tilde{p}_1(i)$ for $i = 1, 2, 3, 4, 5, 6$, where, for example, $\tilde{p}_1(1) = \frac{1}{3}\hat{p}_1(1)$ and $\tilde{p}_1(4) = \frac{4}{7}\hat{p}_1(2)$. Joint probabilities for X_1 and X_2 are denoted by $\tilde{p}(i, j)$ where we have, for example,

$$\tilde{p}(1, j) = \frac{1}{3}\hat{p}(1, j) \quad \text{and} \quad \tilde{p}(4, j) = \frac{4}{7}\hat{p}(2, j).$$

In the more general case, we could “explode” the data at node 1 more generally by replacing instances of 1 with values in $\{1, 2, \dots, \ell_1\}$ some probabilities $q(1, 1), q(1, 2), \dots, q(1, \ell_1)$,

summing to 1, replacing original instances of 2 with values in $\{\ell_1 + 1, \ell_1 + 2, \dots, \ell_1 + \ell_2\}$ with respective probabilities $q(2, 1), q(2, 2), \dots, q(2, \ell_2)$, summing to 1, and so forth.

By “exploding” the data at node one, we have introduced superfluous values for X_1 in terms of probabilities for X_2 . For example, $\tilde{P}(X_2 = j|X_1 = 1) = \tilde{P}(X_2 = j|X_1 = 2)$ for all $j \in \{1, 2, 3\}$. Any proper discretization process should aggregate the values 1 and 2 for X_1 back into one value. Here, $\tilde{P}(X_1 = i, X_2 = j)$ is used to denote the probability $\tilde{p}(i, j)$. Similarly, $\tilde{p}_1(i)$ may be denoted as $\tilde{P}(X_1 = i)$.

For this two-node network, (17) is simply $\hat{I}(X_1, X_2)$, which we will denote by \hat{I} . Define this estimated information as

$$\hat{I} := \sum_{i,j} \hat{I}_{i,j} \quad (18)$$

where

$$\hat{I}_{ij} := \hat{p}(i, j) \cdot \log \left(\frac{\hat{p}(i, j)}{\hat{p}_1(i) \cdot \hat{p}_2(j)} \right) \quad (19)$$

and the sums run over $i \in \{1, 2, 3\}$ and $j \in \{1, 2, 3\}$.

Define the corresponding information \tilde{I} and information terms $\tilde{I}_{i,j}$ using $\tilde{p}(i, j)$ in place of $\hat{p}(i, j)$ with sums running over $i \in \{1, 2, \dots, 6\}$ and $j \in \{1, 2, 3\}$.

It is easy to verify information term relationships such as

$$\tilde{I}_{1,j} = \frac{1}{3} \hat{I}_{1,j} \quad \text{and} \quad \tilde{I}_{4,j} = \frac{4}{7} \cdot \hat{I}_{2,j},$$

and consequently that

$$\tilde{I} = \sum_{\substack{i \in \{1, 2, \dots, 6\} \\ j \in \{1, 2, 3\}}} \tilde{I}_{ij} = \sum_{\substack{i \in \{1, 2, 3\} \\ j \in \{1, 2, 3\}}} \hat{I}_{ij} = \hat{I}.$$

That is, we did not change the information between X_1 and X_2 by exploding the data at X_1 .

We now show that our single threshold top-down search strategy will correctly recover the original values for X_1 . We call this the “correct discretization”, denoted as 12|345|6, which means that we will remove three thresholds from the “full discretization”, denoted as 1|2|3|4|5|6, and that the values 1 and 2 will map back to 1, the values in $\{3, 4, 5\}$ will map back to 2, and the value 6, will map back to 3. We will assume that the original values in $\{1, 2, 3\}$ are all distinct in the sense that $\hat{P}(X_2 = j|X_1 = 1) \neq \hat{P}(X_2 = j|X_1 = 2)$ for some j , $\hat{P}(X_2 = j|X_1 = 1) \neq \hat{P}(X_2 = j|X_1 = 3)$ for some j , and $\hat{P}(X_2 = j|X_1 = 2) \neq \hat{P}(X_2 = j|X_1 = 3)$ for some j , so that they should not be aggregated further.

Removing a Threshold: The Impact on Information

Starting with the exploded data, with values for X_1 represented as $1|2|3|4|5|6$, we consider the (X_1, X_2) information term after removing the threshold between the values r and $r + 1$ for some $r \in \{1, 2, \dots, 5\}$. Note that removal of this r th threshold will leave all values below the threshold unchanged, while all values above will be decreased by 1. For example, if we remove the third threshold, we denote the new configuration as $1|2|34|5|6$, but it represents a mapping

$$\underbrace{1}_{\text{map to 1}} \quad | \quad \underbrace{2}_{\text{map to 2}} \quad | \quad \underbrace{3 \ 4}_{\text{map to 3}} \quad | \quad \underbrace{5}_{\text{map to 4}} \quad | \quad \underbrace{6}_{\text{map to 5}} \quad .$$

Define, for $i \in \{1, 2, \dots, 5\}$ and $j \in \{1, 2, 3\}$, the joint and marginal probabilities for X_1 and X_2 after the r th threshold is removed as $p^{(r)}(i, j)$, $p_1^{(r)}(i)$, and $p_2^{(r)}(j)$. We have, for $j \in \{1, 2, 3\}$, the relationships

$$\begin{aligned} p^{(r)}(i, j) &= \tilde{p}(i, j), & i \in \{1, 2, \dots, r-1\} \quad (r > 1) \\ p^{(r)}(r, j) &= \tilde{p}(r, j) + \tilde{p}(r+1, j) \\ p^{(r)}(i, j) &= \tilde{p}(i+1, j), & i \in \{r+1, r+2, \dots, 6\}, \quad (r < 5) \end{aligned}$$

$$\begin{aligned} p_1^{(r)}(i) &= \tilde{p}_1(i), & i \in \{1, 2, \dots, r-1\} \\ p_1^{(r)}(r) &= \tilde{p}_1(r) + \tilde{p}_1(r+1) \\ p_1^{(r)}(i) &= \tilde{p}_1(i+1), & i \in \{r+1, r+3, \dots, 6\}, \quad (r < 5), \end{aligned}$$

and

$$p_2^{(r)}(j) = \tilde{p}_2(j) = \hat{p}_2(j).$$

Defining $I^{(r)}$ and $I_{ij}^{(r)}$ analogous to (18) and (19), using $p^{(r)}(i, j)$, we have, for $j \in \{1, 2, 3\}$,

$$\begin{aligned} I_{rj}^{(r)} &= p^{(r)}(r, j) \cdot \log \left(\frac{p^{(r)}(r, j)}{p_1^{(r)}(r) p_2^{(r)}(j)} \right) \\ &= [\tilde{p}(r, j) + \tilde{p}(r+1, j)] \cdot \log \left(\frac{\tilde{p}(r, j) + \tilde{p}(r+1, j)}{[\tilde{p}_1(r) + \tilde{p}_1(r+1)] \cdot \tilde{p}_2(j)} \right) \\ &\leq \tilde{p}(r, j) \cdot \log \left(\frac{\tilde{p}(r, j)}{\tilde{p}_1(r) \cdot \tilde{p}_2(j)} \right) + \tilde{p}(r+1, j) \cdot \log \left(\frac{\tilde{p}(r+1, j)}{\tilde{p}_1(r+1) \cdot \tilde{p}_2(j)} \right) \\ &= \tilde{I}_{rj} + \tilde{I}_{r+1, j}. \end{aligned}$$

The inequality is due to the log-sum inequality,

$$\sum_{i=1}^n a_i \log \left(\frac{a_i}{b_i} \right) \geq \left[\sum_i a_i \right] \log \left(\frac{\sum_i a_i}{\sum_i b_i} \right),$$

which holds for any nonnegative a_1, a_2, \dots, a_n and b_1, b_2, \dots, b_n . The log-sum inequality can be shown to be an equality if and only if the a_i/b_i are equal for all $i = 1, 2, \dots, n$. Thus, we have that

$$I_{rj}^{(r)} \leq \tilde{I}_{rj} + \tilde{I}_{r+1,j} \quad (20)$$

for $j \in \{1, 2, 3\}$, with equality if and only if

$$\frac{\tilde{p}(r, j)}{\tilde{p}_1(r) \cdot \tilde{p}_2(j)} = \frac{\tilde{p}(r+1, j)}{\tilde{p}_1(r+1) \cdot \tilde{p}_2(j)}.$$

This happens if and only if

$$\tilde{P}(X_2 = j | X_1 = r) = \tilde{P}(X_2 = j | X_1 = r+1). \quad (21)$$

for $j \in \{1, 2, 3\}$, which is precisely when the values r and $r+1$ in the exploded version of X_1 should be aggregated or discretized into one value. If we do not have (21), aggregating the values will result in a loss of information.

Note that $I^{(r)}$ denotes the value of the information between X_1 and X_2 with the r th threshold in the explosion for X_1 removed. In our example,

$$\tilde{P}(X_2 = j | X_1 = 1) = \frac{\tilde{p}(1, j)}{\tilde{p}_1(1)} = \frac{(1/3)\hat{p}(1, j)}{(1/3)\hat{p}_1(1)} = \hat{P}(X_2 = j | X_1 = 1)$$

and

$$\tilde{P}(X_2 = j | X_1 = 2) = \frac{\tilde{p}(2, j)}{\tilde{p}_2(1)} = \frac{(2/3)\hat{p}(1, j)}{(2/3)\hat{p}_1(1)} = \hat{P}(X_2 = j | X_1 = 1).$$

Thus, we have (21), when $r = 1$, and consequently

$$\begin{aligned} I^{(1)} &= \sum_{i=1}^5 \sum_{j=1}^3 I_{ij}^{(1)} \\ &= \sum_{j=1}^3 I_{1j}^{(1)} + \sum_{i=2}^5 \sum_{j=1}^3 I_{ij}^{(1)} \\ &= \sum_{j=1}^3 I_{1j}^{(1)} + \sum_{i=3}^6 \sum_{j=1}^3 \tilde{I}_{ij} \\ &= \sum_{j=1}^3 (\tilde{I}_{1j} + \tilde{I}_{2j}) + \sum_{i=3}^6 \sum_{j=1}^3 \tilde{I}_{ij} \\ &= \sum_{i=1}^6 \tilde{I}_{ij} = \tilde{I}. \end{aligned}$$

On the other hand, considering the second threshold removal from the full discretization 1|2|3|4|5|6, similar calculations show that

$$\tilde{P}(X_2 = j|X_1 = 2) = \hat{P}(X_2 = j|X_1 = 1)$$

which is not, in general, equal to

$$\tilde{P}(X_2 = j|X_1 = 3) = \hat{P}(X_2 = j|X_1 = 2),$$

so we have the strict inequality $I_{2,j}^{(2)} < \tilde{I}_{2,j} + \tilde{I}_{3j}$ and therefore

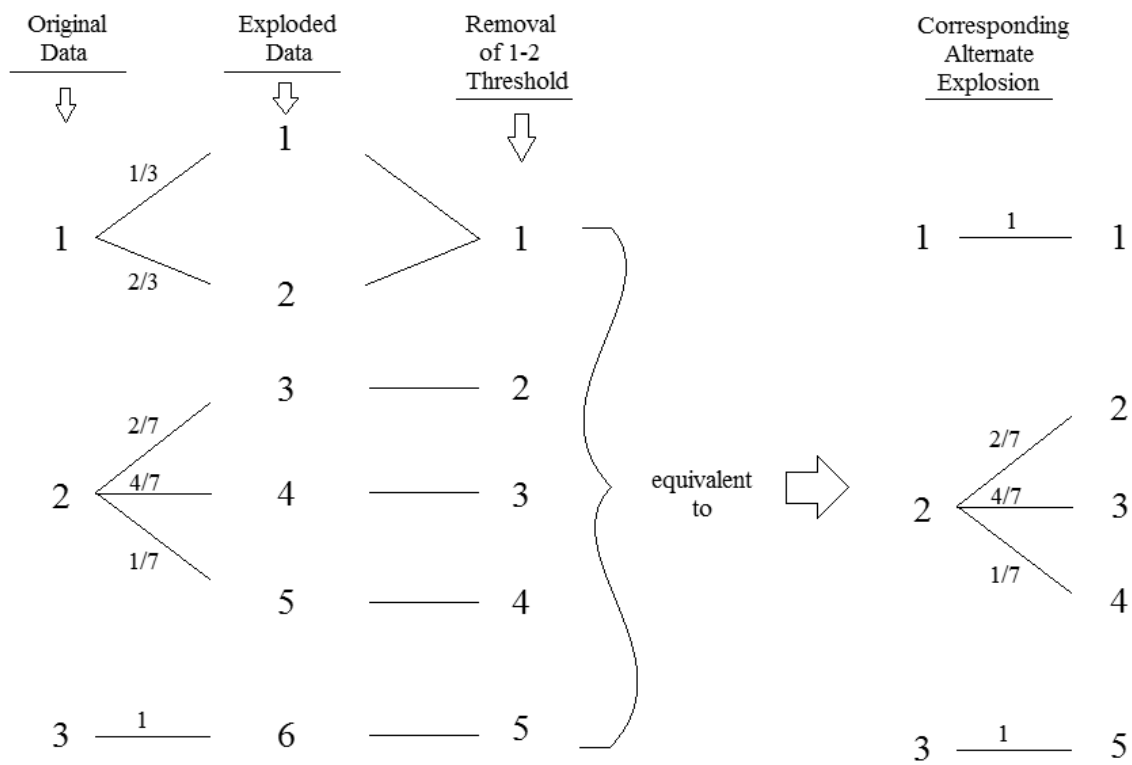
$$\begin{aligned} I^{(2)} &= \sum_{i=1}^5 \sum_{j=1}^3 I_{ij}^{(2)} \\ &= \sum_{j=1}^3 I_{1j}^{(2)} + \sum_{j=1}^3 I_{2j}^{(2)} + \sum_{i=3}^5 \sum_{j=1}^3 I_{ij}^{(2)} \\ &= \sum_{j=1}^3 \tilde{I}_{1j} + \sum_{j=1}^3 I_{2j}^{(2)} + \sum_{i=4}^6 \sum_{j=1}^3 \tilde{I}_{ij} \\ &< \sum_{j=1}^3 \tilde{I}_{1j} + \sum_{j=1}^3 (\tilde{I}_{2j} + \tilde{I}_{3j}) + \sum_{i=4}^6 \sum_{j=1}^3 \tilde{I}_{ij} \\ &= \sum_{i=1}^6 \tilde{I}_{ij} = \tilde{I}. \end{aligned}$$

In all, we can show in this way that

$$I^{(1)} = \tilde{I}, \quad I^{(2)} < \tilde{I}, \quad I^{(3)} = \tilde{I}, \quad I^{(4)} = \tilde{I}, \quad \text{and} \quad I^{(5)} < \tilde{I}. \quad (22)$$

So, our search strategy will produce the correct discretization 12|345|6. The question remains though as to whether this is actually the discretization that minimizes DL_{local} . Due to (20), we will always have $I^{(r)} \leq \tilde{I}$. In fact, due to the log-sum inequality, any removal of a threshold from any configuration with any number of thresholds can never increase information. Thus, the full discretization 1|2|3|4|5|6 will always have maximal information. From (22), we see that the discretizations 12|3|4|5|6, 1|2|34|5|6, and 1|2|3|45|6 have the same, and thus also maximal, information. Note that each of these three configurations corresponds to a different explosion of the original data. This is illustrated for 12|3|4|5|6 in Figure 3. Thus, by the same information arguments above, “correct removal” (removal of one of the superfluous thresholds) of a threshold from 12|3|4|5|6 will result in another configuration with the same maximal information. Since this is true starting with any one of the three correct single threshold removal configurations, we see that we can remove two of the superfluous thresholds from the full explosion and still maintain the maximal information. Continuing this argument, we can remove all superfluous thresholds from the full discretization and the resulting configurations of thresholds, which in this example is 12|345|6, will have the maximal information.

Figure 3: Correct Removal of a Threshold Corresponds to an Alternate Explosion



5.2 A Closer look at the Leading Terms in DL_{local}

For a general DAG, the leading terms in DL_{local} , given in (16) can be rewritten as

$$(m_1 - 1)H\left(\frac{k_1 - 1}{m_1 - 1}\right) + \log k_1 + \frac{1}{2}(\log m)(ck_1 - \|\Pi_1\|) =: D(k_1)$$

for some $c \geq 0$ where $c = 0$ if and only if node 1 is not connected to any other nodes.

The second and third terms here are clearly increasing in k_1 . The first term, as a function of k_1 , is symmetric about $k_1 = (m_1 + 1)/2$, increasing to the left of this value and decreasing to the right. However, since

$$\left.\frac{d}{dx}D(x)\right|_{x=k_1} = \log\left(\frac{m_1 - k_1}{k_1 - 1}\right) + \frac{1}{k_1} + \frac{1}{2}c \cdot \log m,$$

we can, when $c > 0$ choose m (the sample size) large enough to ensure that $D(k_1)$ is increasing in k_1 . In this case, this first part of DL_{local} acts as a penalty term and DL_{local} is minimized by choosing the discretization that maximizes the information terms with the minimum number of thresholds. By design, the single threshold top-down search strategy will do exactly this. Thus, when employing the strategy for large samples, we can ignore these leading terms.

5.3 More Complicated Networks

The observations in Section 5.2 were not dependent on the specific network, however, our analysis of the information terms was for a two-node network where X_1 was a parent to X_2 . Since $\hat{I}(X_1, X_2) = \hat{I}(X_2, X_1)$, our search strategy will also find the “optimal discretization” for X_1 , i.e. the one that maximizes (17) with a minimum number of thresholds, for the two-node network where X_1 is a child of X_2 .

We will now check that the strategy will find the optimal discretization for X_1 for general networks. To this end, we begin by independently considering the two types of terms in (17).

- $\hat{I}(X_1, \Pi_1)$

Since replacing X_2 with a vector of random variables has no effect on any of the computations in Section 5.1, we see that our search strategy will find the discretization for X_1 that maximizes $\hat{I}(X_1, \Pi_1)$ with a minimum number of thresholds.

- $\hat{I}(X_j, \Pi_j)$ where $X_1 \in \Pi_j$

Suppose, for ease of exposition, that $j = 2$ and, that all nodes originally take values in $\{1, 2, 3\}$, and that X_1 has been exploded to take values in $\{1, 2, 3, 4, 5, 6\}$ just as in, and using the same probabilities as, Section 5.1. If Π_2 consists only of X_1 , we have already seen that our search strategy will maximize the information term with a minimum number of thresholds. Assuming now that Π_2 consists of X_1 and some vector \vec{Y} whose components are the other parents of X_2 , we have

$$\hat{I} := \hat{I}(X_2, \Pi_2) = \hat{I}(X_2, (X_1, \vec{Y})) = \sum_{ij\vec{k}} \hat{I}_{ij\vec{k}}$$

where

$$\hat{I}_{ij\vec{k}} = \hat{p}(i, j, \vec{k}) \cdot \log \left(\frac{\hat{p}(i, j, \vec{k})}{\hat{p}_2(j) \hat{p}_{1Y}(i, \vec{k})} \right).$$

Here, $\hat{p}(i, j, \vec{k}) = \hat{P}(X_1 = i, X_2 = j, \vec{Y} = \vec{k})$, $\hat{p}_2(j) = \hat{P}(X_2 = j)$, and $\hat{p}_{1Y}(i, \vec{k}) = \hat{P}(X_1 = i, \vec{Y} = \vec{k})$. Using \tilde{p} to denote probabilities after the explosion of X_1 , it is easy to see expected relationships such as

$$\tilde{p}(4, j, \vec{k}) = \frac{4}{7} \hat{p}(2, j, \vec{k}), \quad \tilde{p}_2(j) = \hat{p}_2(j), \quad \text{and} \quad \tilde{p}_{1Y}(2, \vec{k}) = \frac{2}{3} \hat{p}_{1Y}(1, \vec{k}).$$

Therefore, we can verify, for example, that $\tilde{I}_{2j\vec{k}} = (2/3) \hat{I}_{1j\vec{k}}$, and that the overall information terms \tilde{I} and \hat{I} are equal. We can also verify, using the log-sum inequality, that after removal of the r th threshold we have

$$I_{r,j,\vec{k}}^{(r)} \leq \tilde{I}_{r,j,\vec{k}} + \tilde{I}_{r+1,j,\vec{k}},$$

with equality if and only if

$$\tilde{P}(X_2 = j | X_1 = r, \vec{Y} = \vec{k}) = \tilde{P}(X_2 = j | X_1 = r + 1, \vec{Y} = \vec{k}). \quad (23)$$

Summing over all appropriate values for indices, we get that $I^{(r)}$, the information term between X_2 and its parents after removal of the r th threshold for X_1 , is less than or equal to \tilde{I} , with equality if and only if (23) holds for all j and \vec{k} . When $r = 1$, for example, both sides of (23) are equal to $\hat{P}(X_2 = j | X_1 = 1, \vec{Y} = \vec{k})$, indicating that the first threshold should be removed. When $r = 2$, the left side is equal to $\hat{P}(X_2 = j | X_1 = 2, \vec{Y} = \vec{k})$ and the right side is equal to $\hat{P}(X_2 = j | X_1 = 3, \vec{Y} = \vec{k})$, so we do not have (23) and hence a decrease in information. Thus, we would not remove the second threshold.

\tilde{I} , the information with all thresholds in place, is the maximal information among all possible discretizations. If we remove each threshold that leaves the information unchanged, we will still have the maximal information with a minimum number of thresholds. Thus, the top-down search strategy will give the optimal discretization.

Considering all terms in (17) independently may result in different discretizations. For example, consider graph 8 from Table 1. In this case, (17) becomes $\hat{I}(X_1, X_2) + \hat{I}(X_1, X_3)$. If $\hat{P}(X_2 = j|X_1 = 1) = \hat{P}(X_2 = j|X_1 = 2)$ for all j but $\hat{P}(X_3 = j|X_1 = 1) \neq \hat{P}(X_3 = j|X_1 = 2)$ for some j , removing the threshold between 1 and 2 for the discretization of X_1 would leave the information between X_1 and X_2 unchanged but would decrease the information between X_1 and X_3 . However, as both information terms, and hence their sum, are maximized with the full discretization for X_1 , the top down-search strategy will not allow us to remove the threshold between 1 and 2. That is, it will only remove thresholds between values of X_1 that are indistinguishable in terms of the conditional distributions involving all nodes connected to X_1 .

6 Conclusions

We have seen, in the case of ideal “exploded” data where there is a “correct” discretization, that the minimum description length scoring mechanism of Friedman and Goldszmidt will in fact recover the discretization. Just as importantly, we have seen that we can find it from among 2^{m_1-1} possibilities by making only $m_1 - 1$ comparisons.

In the case of discrete data where superfluous values were not manufactured, for example the two-node network where X_1 is a parent to X_2 that originally takes values in $\{1, 2, 3, 4, 5, 6\}$ and $\{1, 2, 3\}$, respectively, we should aggregate 1 and 2 for node 1 into a single value if $P(X_2 = j|X_1 = 1) = P(X_2 = j|X_1 = 2)$ for all $j \in \{1, 2, 3\}$. From the data, we will only get to see that $\hat{P}(X_2 = j|X_1 = 1) \approx \hat{P}(X_2 = j|X_1 = 2)$. Even with a large sample size, because of the approximation, we would still see some decrease in overall information when correctly removing the threshold, so it remains to determine when such a decrease is significant. We have much empirical evidence that we will still be able to recover the correct discretization by comparing DL_{local} for only single threshold removals to DL_{local} for the full discretization. As an example, we simulated 100,000 values for X_1 and X_2 in the two-node network by simulating X_1 in $\{1, 2, 3, 4, 5, 6\}$ directly (as opposed to first simulating them in $\{1, 2, 3\}$ and then exploding the data). We chose parameters such that

$P(X_2 = j|X_1 = 1) = P(X_2 = j|X_1 = 2)$ and $P(X_2 = j|X_1 = 5) = P(X_2 = j|X_1 = 6)$ for all $j \in \{1, 2, 3\}$. In Table 3, we show the full discretization score, all single threshold removal scores, and the true discretization score. The two incorrect threshold removals (between 1 and 2 and between 5 and 6) stand out as having different DL_{local} scores than the rest. The quantification of this difference is still ongoing work.

Table 3: Local Description Length Score

Discretization	DL local
1 2 3 4 5 6	-29841.52
12 3 4 5 6	-29870.53
1 23 4 5 6	-24456.07
1 2 34 5 6	-29866.02
1 2 3 45 6	-29896.74
1 2 3 4 56	-24585.90
12 345 6	-29929.17

Computation of the DL_{local} score requires that one have a graph under consideration. Indeed, all of our arguments in this paper have been made using the correct generating graph. In practice, it seems that we can still recover the correct discretization using **any** graph as long as the node to be discretized is connected to at least one other node which, incidentally, is precisely when we can ignore the leading terms in DL_{local} as discussed in Section 5.2. We recommend running the search for the discretization that minimized DL_{local} on several networks. Fortunately, our proposed fast search strategy makes this feasible.

The case of truly continuous data is still problematic. For the approach of this paper, it is incorrect to use estimates of densities and information terms for continuous random variables, as the procedures discussed all involve comparisons of scores computed after proposed discretizations. Consider again the two node network where X_1 is a parent of X_2 and node 1 is the one to be discretized. (Recall that the MDL discretization process is for one node at a time. It is assumed that other nodes have already been discretized and that they will be “re-discretized” pending discretization of the current node in a cyclic fashion.) Relabel the $m_1 = m$ distinct values observed for X_1 as 1 through m . We should aggregate 1 and 2 for node 1 into a single value if $\hat{P}(X_2 = j|X_1 = 1) \approx \hat{P}(X_2 = j|X_1 = 2)$ for all $j \in \{1, 2, 3\}$, however approximations of such probabilities will can only be 0 or 1! Perhaps

then, a bottom-up approach to minimizing DL_{local} would be better though we have not, as yet, had great success with simulated results. This too is ongoing work.

References

- [1] H. Akaike. Information theory and an extension of the maximum likelihood principle. In B.N. Petrov and F. Csaki, editors, *Second international symposium on information theory*, pages 267–281. Budapest: Academiai Kiado, 1973.
- [2] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.
- [3] H. Akaike. Likelihood of a model and information criteria. *Journal of Econometrics*, 16:3–14, 1981.
- [4] J. Bockhorst, M. Craven, D. Page, J. Shavlik, and J. Glasner. A Bayesian network approach to operon prediction. *Bioinformatics*, 19:1227–1235, 2003.
- [5] H. Bozdogan. Akaike’s information criterion and recent developments in information complexity. *Journal of Mathematical Psychology*, 44:62–91, 2000.
- [6] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Hoboken, New Jersey, 2006.
- [7] R.G. Cowell, R.J. Verall, and Y.K. Yoon. Modeling operational risk with Bayesian networks. *Journal of Risk and Insurance*, 74:795–827, 2007.
- [8] D.I. Curiac, G. Vasile, O. Baniias, C Volosencu, , and A. Albu. Bayesian network model for diagnosis of psychiatric diseases. In *Proceedings of the ITI 2009 31st International Conference on Information Technology Interfaces*, pages 61–66. 2009.
- [9] C.O. Daub, R. Steur, J. Selbig, and S. Kloska. Estimating mutual information using B-spline functions an improved similarity measure for analysing gene expression data. *BMC Bioinformatics*, 5:118:1471–2105, 2004.
- [10] N. Fenton and M. Neil. *Risk Assessment and Decision Analysis with Bayesian Networks*. CRC Press, Taylor & Francis Group, Boca Raton, FL, 2013.
- [11] N. Friedman and M. Goldszmidt. Discretizing continuous attributes while learning Bayesian networks. In *Proceedings of ICML-1996*, pages 157–165.

- [12] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, 7:601–620, 2000.
- [13] T.L. Griffiths, C. Kemp, and J.B. Tenenbaum. Bayesian models of cognition. In R. Son, editor, *Cambridge Handbook of Computational Cognitive Modeling*, pages 59–100. Cambridge University Press, 2008.
- [14] R.L. Kashvap. A Bayesian comparison of different classes of dynamic models using empirical data. *IEEE Transactions on Automatic Control*, 22(5):715–727, 1977.
- [15] K. Korb and A. Nicholson. *Bayesian Artificial Intelligence*. Chapman and Hall/CRC, Boca Raton, FL, 2003.
- [16] P. Larrañaga, B. Calvo, R. Santana, C. Bielza, J. Galdiano, I. Inza, J. Lozano, G. Santafè R. Armañanzas, A. Pèrez, and V. Robles. Machine learning in bioinformatics. *Briefings in Bioinformatics*, 7(1):86–112, 2005.
- [17] R. Mourad, C. Sinoquet, and P. Leray. A hierarchical bayesian network approach for linkage disequilibrium modeling and data-dimensionality reduction prior to genome-wide association studies. *BMC Bioinformatics*, 12(1):16, 2011.
- [18] A. Onisko, M.J. Druzdzel, and H. Wasyluk. A Bayesian network model for diagnosis of liver disorders. *Proceedings of the Eleventh Conference on Biocybernetics and Biomedical Engineering*, 2:842–846, 1999.
- [19] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–658, 1978.
- [20] G. Schwartz. Estimating the dimension of a model. *The Annals of Statistics*, 5(2):461–464, 1978.
- [21] J.B. Tenenbaum, T.L. Griffiths, and C. Kemp. Theory-based Bayesian models of inductive learning and reasoning. *Trends in Cognitive Sciences*, 10(7):309–318, 2006.
- [22] P. Weber, G. Medina-Oliva, C. Simon, and B. Iung. Overview on Bayesian networks applications for dependability, risk analysis and maintenance areas. *Engineering Applications of Artificial Intelligence*, 25(4):671–682, 2012.
- [23] Y. Xiang, B. Pant, A. Eisen, M.P. Beddoes, and D. Pool. Multiply sectioned Bayesian networks for neuromuscular diagnosis. *Artificial Intelligence in Medicine*, 5:293–314, 1993.

